Problem Set #5

Greg Plaxton

April 8, 2019

Your solutions are required to be typeset using LaTeX, and to be turned in electronically using Canvas. As discussed on the course syllabus, there are two deadlines for this homework. The first deadline is 8pm on Friday, April 19. The second deadline is 8pm on Sunday, April 21. Submissions received before the first deadline will not be charged any slip days. No submissions will be allowed after the second deadline. See the syllabus for further details regarding the slip day policy.

1. Let $G = (V, E)$ be a flow network, and for any edge e in E, let $c(e)$ denote the capacity of edge e. Consider the following linear program defined with respect to G. There is a variable α_e for each edge e in E, and a variable δ_v for each vertex v in V. The objective is to minimize $\sum_{e \in E} \alpha_e c(e)$ subject to the following constraints: (1) $\alpha_{(u,v)} + \delta_v - \delta_u \geq 0$ for all (u, v) in E; (2) $\alpha_e \geq 0$ for all e in E; (3) $0 \leq \delta_v \leq 1$ for all v in V; (4) $\delta_s = 1$; (5) $\delta_t = 0$. Prove that this linear program admits a 0-1 optimal solution.

2. On the lecture slides related to linear programming, we presented an LP formulation of the minimum-cost perfect matching problem. Let I be a feasible instance of this linear program. Use Hall's theorem (see Exercise 26.3–4 on page 735) to prove that I admits a 0-1 optimal solution.

3. [Remark: Please remember that for any clause C appearing in a 3-CNF formula, we require the variables associated with the three literals in C to be distinct.] Let us define a 3-CNF formula f to be *balanced* if each variable in f appears (whether negated or not) in exactly three clauses. Prove that the restricted version of 3-SAT in which the input 3-CNF formula is required to be balanced belongs to P. Hint: Make use of the result of Exercise 26.3–5 on page 736.

4. Let us define a $(2, 3)$-CNF formula as a propositional formula that is the conjunction of a number of clauses, each of which is the disjunction of either 2 or 3 literals, where no two literals in the same clause are associated with the same variable. We define $(2, 3)$-SAT as the problem of determining whether a given $(2, 3)$-CNF formula is satisfiable. Given that 3-SAT is NP-complete, it is easy to see that $(2, 3)$-SAT is NP-complete. Let us define a $(2, 3)$-CNF formula to be *nice* if no literal appears in more than two clauses.

(In other words, for any variable x, the literal x appears in at most two clauses, and the literal $\neg x$ appears in at most two clauses.) Prove that the restricted version of $(2, 3)$-SAT in which the input is required to be nice is NP-complete.